PDG Computing Review

The RPP Programs

Outline

- Overview
- Sample Program descriptions
  - The fit program
  - The printr program
- Summary

Overview

- Programs and utilities are used to generate the RPP book from the data in the database.
  - 24 programs do most of the work of generating the book
  - 30 utilities (shell scripts that use the database) have a minor role and will not be discussed further.

Program functions

- Programs serve several different functions
  - Producing RPP in TeX format. (10 programs)
  - Producing plots for RPP (4 programs)
  - Making calculations and updating the database (4 programs)
  - Monitoring the book production (4 programs)
  - Other programs (2 programs)

Program structure

- Programming language
  - Programs are mostly written in Fortran
  - A few subroutines are written in C but are Fortran callable.
  - Everything compiles and runs without modification on most platforms including SunOS and Linux
- Lines of code
  - Common librarys 28k active code 25k comments
  - Individual programs 28k active code 13k comments

Database connection

- Programs use SQL and mostly do not care which database engine is used
  - Branch on database engine in about 20 places.
- Programs connect to the database using a small library
  - Written in C but Fortran callable
  - Have versions for Oracle and for PostgreSQL
  - Easy to add another database engine
- Lines of code
  - Oracle 900 active code 500 comments
  - PostgreSQL 1200 active code 500 comments

The fit program

- Does a constrained fit to a group of overdetermined nodes
- Overall structure
  - Get list of available fits and decide which fits to perform.
  - Process each fit that is to be performed
    - Get input data from the database
    - Do the least squares fit (no database involvement here)
    - Put results into the database.

Fit – determine which fits to perform

- Get list of possible fits and their algorithm from FIT_CONTROL1 table
- Ask user which fits to perform.

Fit – get input data (I)

- Get type of algorithm from FIT_CONTROL1 table.
  - 'BR' Branching ratio (must sum to one)
  - 'MASS' Mass fit (no summation requirement)
  - 'IGNORE' Do not process the fit
- Get all nodes that participate in the fit from FIT_CONTROL2 table
- Determine type of data from the TREE table.
- Get initial values for each node from the FIT_SEED table.

Fit – get input data (II)

- Get relations between the nodes from the RELATIONSHIP table
- For each node get all the measurements from the MEASUREMENT table.
- Get correlations between measurements of different nodes from the CORRELATION table.

Fit – do the actual least squares fit

- Do a basic least squares fit to determine the value of the parameters and nodes adjusting the input errors for measurements with asymmetric errors
- Discard any very poor measurements and refit
- For each measurement with asymmetric errors move the measurement by one standard deviation and refit to determine the errors on the fitted parameters
- Refit to calculate the scale factors for the errors

Fit – Update database with results

- Delete old values for RESULT_SUMMARY and FIT_CORRELATION_MATRIX tables
- For each node, insert new values into the RESULT_SUMMARY table
- Update general information in the FIT_CONTROL1 table
  - Number of parameters
  - Number of measurements
  - Chi square
- Update correlation matrix in the FIT_CORRELATION_MATRIX table

The printr program

- Generates the listings (input to TeX)
- Overall structure
  - Get the starting and ending nodes from the user
  - Traverse the tree table to get a list of all the nodes to be process in the order in which they are to be processed
  - Process each node.

Printr – processing a node

- Write heading lines for the beginning of banners and particles
- For data nodes
  - Write a header for the node
  - Process the data block
  - Leave space for an ideogram if one is to be included
- For partial mean life or partial decay nodes
  - Generate a table of partial decay modes
  - Write information about any fits involving the node
- At the end of a particle write out footnotes and references

PDG listings – Lambda mass

NODE=S018

Λ

$I(J^P) = 0(\frac{1}{2}^+)$ Status: ****

We have omitted some results that have been superseded by later experiments. See our earlier editions.

NODE=S018

Λ MASS

NODE=S018205

The fit uses Λ, Σ^+, Σ^0, Σ^- mass and mass-difference measurements.

NODE=S018205

NODE=S018M

VALUE (MeV)	EVTS	DOCUMENT ID		TECN	COMMENT
1115.683 ± 0.006 OUR FIT					
1115.683 ± 0.006 OUR AVERAGE					
1115.678 ± 0.006 ± 0.006	20k	HARTOUNI	94	SPEC	*pp* 27.5 GeV/*c*
1115.690 ± 0.008 ± 0.006	18k	[1] HARTOUNI	94	SPEC	*pp* 27.5 GeV/*c*
• • • We do not use the following data for averages, fits, limits, etc. • • •					
1115.59 ± 0.08	935	HYMAN	72	HEBC	
1115.39 ± 0.12	195	MAYEUR	67	EMUL	
1115.6 ± 0.4		LONDON	66	HBC	
1115.65 ± 0.07	488	[2] SCHMIDT	65	HBC	
1115.44 ± 0.12		[3] BHOWMIK	63	RVUE	

OCCUR=2

[1] We assume *CPT* invariance: this is the $\bar{\Lambda}$ mass as measured by HARTOUNI 94. See below for the fractional mass difference, testing *CPT*.

NODE=S018M;LINKAGE=C

[2] The SCHMIDT 65 masses have been reevaluated using our April 1973 proton and $K^\pm$ and $\pi^\pm$ masses. P. Schmidt, private communication (1974).

NODE=S018M;LINKAGE=A

[3] The mass has been raised 35 keV to take into account a 46 keV increase in the proton mass and an 11 keV decrease in the $\pi^\pm$ mass (note added Reviews of Modern Physics **39** 1 (1967)).

NODE=S018M;LINKAGE=L

Printr – processing a data node

- Get each measurement from the MEASUREMENT table to determine what columns of information need to be written
- Get alignment from the COLUMN_HEADER or use standard alignment to determine the position of each of the columns
- Build the column headers
- Get, align, typeset and write the summarys (fit and fit and average values)
- Get, align, typeset, and write the measurements. Order “used” before “not used”

PDG Listings – Lambda decay modes

Λ DECAY MODES

NODE=S018235;NODE=S018

	Mode	Fraction (Γ_i/Γ)	
Γ_1	$p\pi^-$	(63.9 ±0.5) %	DESIG=1
Γ_2	$n\pi^0$	(35.8 ±0.5) %	DESIG=2
Γ_3	$n\gamma$	(1.75±0.15) × 10^{-3}	DESIG=6
Γ_4	$p\pi^-\gamma$	[a] (8.4 ±1.4) × 10^{-4}	DESIG=5
Γ_5	$pe^-\overline{\nu}_e$	(8.32±0.14) × 10^{-4}	DESIG=4
Γ_6	$p\mu^-\overline{\nu}_\mu$	(1.57±0.35) × 10^{-4}	DESIG=3

[a] See the Listings below for the pion momentum range used in this measurement.

LINKAGE=SD

Printr- generate a table of partial decay modes

- Get alignment information for the decay modes
- Get tabular information from the COLUMN_HEADER table or use standard tabular alignment
- Format and write the column headers
- Get the partial decay modes from the DECAY table, find and write their best values, and save any decay table footnotes
- When done append any decay table footnotes

PDG listings – Lambda fit information

CONSTRAINED FIT INFORMATION

An overall fit to 5 branching ratios uses 20 measurements and one constraint to determine 5 parameters. The overall fit has a $\chi^2 = 10.5$ for 16 degrees of freedom.

The following *off-diagonal* array elements are the correlation coefficients $\langle \delta x_i \delta x_j \rangle / (\delta x_i \cdot \delta x_j)$, in percent, from the fit to the branching fractions, $x_i \equiv \Gamma_i / \Gamma_{\text{total}}$. The fit constrains the x_i whose labels appear in this array to sum to one.

x_2	−100			
x_3	−2	−1		
x_5	46	−46	−1	
x_6	0	0	0	0
	x_1	x_2	x_3	x_5

Printr- write information about fits involving the node

- Generate a paragraph describing the fit
  - Get the type and number of each type of nodes involved in the fit. Use names for each type to put into the paragraph
- Format and write the correlation matrix for the fit

Summary

- Programs are well structured and reasonably well documented.
- They are complex since they impliment complex algorithms.
- I am the only person that is initmately familiar with the programs
  - I am currently retired.